Blockchain Anchoring Audit Record

Complete cryptographic timestamp register, 2019 to 2026

Field	Value
Document reference	IIC-BAR-2026-08-30
Version	2.0
Date of issue	30 August 2026
Supersedes	IIC-BAR-2026-06-14 v1.0, issued 14 June 2026
Issued by	Https Card - Internet Identity Card Ltd
Company number	09168431, England and Wales
Registered office	124 City Road, London EC1V 2NX, United Kingdom
Author	Michael Benaudis
Anchors recorded	29, all resolved and confirmed on chain
Period covered	13 May 2019 to 30 August 2026
Classification	Confidential - for audit use

Statement of authority. Every entry in this record was resolved directly against the Bitcoin and Ethereum blockchains on 30 August 2026, using a public block explorer API and a public JSON-RPC endpoint respectively. No timestamp in this document is taken from a certificate display; every time given is a block time read from the chain and expressed in UTC. The reader is not asked to trust this document: section 7 gives the procedure to reproduce every claim independently.

Correction notice. Version 1.0 of this record reported four anchor times in Central European Summer Time under a column headed "Anchored at (UTC)". All four were two hours ahead of the true block time. Version 2.0 corrects them and records the cause in section 8. The transaction identifiers and calendar dates in version 1.0 were correct.

1. Document control

Revision history

Version	Date	Summary of changes
1.0	14 Jun 2026	Initial issue. Four anchors documented (2019 IPFS release; three June 2026 dual-chain anchors). Anchor times reported in CEST under a UTC label.
2.0	30 Aug 2026	Complete rebuild. 29 anchors instead of 7 transactions. Every block time re-read from the chain in UTC, correcting the four CEST values of v1.0. Register extended from 17 June to 30 August 2026. Verification procedures added for both anchoring schemes, neither of which v1.0 documented. Merkle inclusion proofs published. Two anchors present in v1.0 but absent from the public site are now published there.

2. Executive summary

This record documents every blockchain anchor created by Https Card - Internet Identity Card Ltd between May 2019 and August 2026. It exists so that any third party can establish, without trusting the company or any service it uses, that a given artefact existed in a given form at a given time.

29 transactions are recorded. All 29 were resolved on chain and are confirmed. No reference in this document is unverified, malformed or unconfirmed.

The record has one solo anchor, from 2019, and fourteen Bitcoin/Ethereum pairs. Every release since May 2026 is anchored on both chains, so a verifier who distrusts one chain or its operators may rely on the other.

A transaction identifier does not identify a release. The timestamping service aggregates submissions into one Merkle tree per chain per window, and commits only the tree root. One transaction may therefore cover several unrelated documents, and one document may appear under a transaction that also covers others. The unit of proof is the leaf together with its inclusion path, not the transaction. Three anchors in this register carry more than one leaf; entry 22 carries three. This is stated plainly because reading the register as one transaction per release would be wrong.

Two distinct anchoring schemes appear. The 2019 anchor derives a Bitcoin address from the Merkle root and funds it; the root is not written on chain and must be reconstructed. Every 2026 anchor writes the root directly, in a Bitcoin OP_RETURN or in Ethereum calldata. Both are documented in section 7; the second is materially easier to check.

3. Method

3.1 How an anchor is created

At each release the issuer computes the SHA-256 digest of every artefact and submits those digests to a blockchain timestamping service. The files themselves are never transmitted. The service aggregates the digests received in a time window into a Merkle tree and commits the tree root to Bitcoin and to Ethereum. The block time of the committing transaction is the timestamp.

3.2 What an anchor proves, and what it does not

An anchor establishes	An anchor does not establish
That a file with this exact digest existed no later than the block time.	That the file did not exist earlier. An anchor is an upper bound on a date of existence, never a lower bound.
That the file has not changed since, if the digest still matches.	Any statement about the content, accuracy or legal effect of the file.
That no party, the issuer included, can substitute the artefact retroactively.	Authorship. An anchor binds a digest to a time, not to a person.

3.3 Why two chains

Bitcoin and Ethereum rest on different consensus mechanisms and different operator sets. Anchoring to both means a verifier who rejects one may still rely on the other. The scheme is agnostic to the choice: any append-only public ledger would serve.

3.4 Time

Every time in this document is the UTC block time of the committing transaction, read from the chain. Certificate displays are not used. Every timestamping certificate examined in preparing this version showed local time, two hours ahead of the block time, under a UTC-looking label. Ten instances were checked and all ten carried the same offset. Section 8 records the consequences.

4. Chronological register

Ordered by block time. BTC blocks resolved through the mempool.space API, ETH blocks through a public JSON-RPC endpoint, both on 30 August 2026. "Committed root" is the Merkle root actually written on chain, read from the transaction itself and not from any certificate.

#	Block time (UTC)	Chain	Block	Transaction and committed Merkle root
1	2019-05-13 00:13:27	BTC	575 758	`929bb2d8d5230c372f145d40e14d4c64ff70ef04b6dd389fc5bc7342fc754540` `not written on chain - derived-address scheme, see 7.1`
2	2026-05-21 18:06:59	ETH	25 145 310	`0x589165f14d9fb3a3ea9be7fda7e3925b0fccfb09a29e24685cc4fdd16d53152b` `5011e34ec7c60fbcb5a8c0130098336d6cd62796daa080fd4ad5e2f9d71a65b5`
3	2026-05-22 00:15:56	BTC	950 432	`48c7e6959a3c3933baa4bde43c4c55038234de5c26fbad3ca70e8ccc1d400a8c` `ed744de180567be47ccb06f71bf3e5d80668f1b53af83b421c483f9b1255b2a1`
4	2026-06-07 00:12:14	BTC	952 666	`ed64e62720f6cbdb883f749fd7824b7a19858cacd722821ef1c936c10148688b` `97903ae0fc81c02b564b89303f492d2c79b1cb1a1831bacca66b23205d564219`
5	2026-06-07 06:07:11	ETH	25 263 622	`0xf2393b1c0ab6403fb5391199c7550c71f6f86b56acf492e920b207b74cc8afa9` `3a45728332d5c57b9aab4bd9cb2cf19db075eb5d8f6a1a16d91640491aa109dc`
6	2026-06-12 00:22:00	BTC	953 295	`7c3ebcf1c429cdadd681d0ea77841bf66a43c4d70709d422bcd5a6f25500d11a` `2a05bfc48f09805fd4a7c9a4d76f6e6bc786d08857b9ad4b80d1d6ab0658d534`
7	2026-06-12 06:07:23	ETH	25 299 490	`0xca22f41a6caf54d403ecf1eb8c705ae6b9ee543de5346c8dd40ab8f1efeca313` `bd80158c14b360849cdaebc9959c4610b2366a2e45b5726ac177b2bcb002f408`
8	2026-06-17 00:32:47	BTC	954 025	`79391878831835085c30cdcd512f563fc9fa59a9c93befdb063834096acb7f9c` `e130f8ca26a8bf4d387434bef3acdb77d4699ba13f35d3f5b4827e00417e5e80`
9	2026-06-17 06:06:47	ETH	25 335 365	`0x0f0ac02a4c1358bdb66a73935fd6221a06f342a8c917bb32c6514af95ede5ecb` `94028c6fd6c24298c991977ae6b93a0482b1a266c59ae562af7a3627826d28d4`
10	2026-06-18 18:06:47	ETH	25 346 133	`0x2a2da35b2ade7cd63bbe0591e65414610f2ca28437f8d585714ca37a5700f3f7` `36596f7008883ea6f728cd9625acb280cbe6e57fc1b36305e496375022ebc07c`
11	2026-06-19 00:09:14	BTC	954 321	`032aacc2df6bddc473173c154c90fe6175456441688d0cc755a21caacd148cc3` `70a239bf0b2d170f1cbd03de9be532065f2160b1384c719d7c1e2c81ced3bb93`
12	2026-07-03 00:16:28	BTC	956 416	`9ce8abafacba4591f1f95ca9ae2805aa6e3b3055c638a479f8c1a6e6a568fc4d` `49b7bd41493e0df6f808cbc3ac4078a59af28c6acc2bd96aabe90c8e48b0844b`
13	2026-07-03 06:07:23	ETH	25 450 092	`0x21322a64c801eda7e7d1f9fd94acb8dcc0b715be9dbcd57a2d6f75c34a5397c8` `eed3bfef4e25b3fc20432ee919037751a584d4602bdcfbf6a2a23bb3c94e1d47`
14	2026-07-04 18:06:47	ETH	25 460 843	`0xe95909432302d32f2919bc797a00a4a9d9b00f68d1f60b3cfb11b13b35dee58f` `760bbe04d9aa0f58f47e1e3f6bf82b17cecadbcafa7eecfe95692646972ffaf8`
15	2026-07-05 00:15:20	BTC	956 720	`34ce2121346840a4d5cbfea2c562a118b268f4541e932348478754351add66bf` `a76ec7949e2d450fe654c52c9acb97f2b8a1024f85dcec8b290c053d12c4784d`
16	2026-07-23 18:06:47	ETH	25 597 114	`0xa83878c13f76dde7d1fb41e0dd448f50435b4a4d74c818ff7095271bff787156` `378f7ae730ebc4aff03cf2e179d28235a0c2289593ae671dae9c699abc66cad9`
17	2026-07-24 00:15:25	BTC	959 329	`1d4372d3949227dedf57131d708f14f065037e52f643add943e7a729aa8f9ca7` `3e2127254a984302799613829 41a9018604bc46e7477a23cbeda961005eb227e`
18	2026-08-07 00:30:52	BTC	961 351	`30658ff6895e6a0ca02cc4679e5c09118be240665bbddc5f0221c592bddc0b81` `86afb1b7ce248b18b126c8aa079a973f10260d8b4ea01e2ecd6665792cd559d6`
19	2026-08-07 06:06:47	ETH	25 701 188	`0xc569312fc4a47d3943580131649bfbd1a6647ca1d3c1fd022118638c38aaa96a` `793966c9f6fc4f7e31404d260110207c3a2302dbf3bae61f3c23fd1d7a191011`
20	2026-08-07 18:06:47	ETH	25 704 773	`0x049958f761e2af0627d85bb413e43a6726fd7830988c5c839d56b8dabe328e7f` `91552fc83818ffbc1685e59bb9e47f43dea2a66716592251937dce40a7384d3b`
21	2026-08-08 00:36:47	BTC	961 512	`4cdc281d65e856db22a9ab3a80335a43a68ec1d62cc998f09497b17855872a5e` `02d6cc4f01295c0a9be92c555c7f466dd5a87ab950efac6439a09e2b1477ae9d`
22	2026-08-24 12:22:05	BTC	963 856	`f5ad2a35f5d5c63598b66f6f25f4b6970f70a3287debf89d6d0512b374abd60b` `6f6ddc8eb8619b4ee7693e6d275ef8149ce4792cee6499b33d025c8f6cef423c`

#	Block time (UTC)	Chain	Block	Transaction and committed Merkle root
23	2026-08-24 18:06:47	ETH	25 826 727	`0x550c8c3355d7c5041a34d97c6fa7f1dcfa03c7a48d3ae4e58a98ad74c7997bf5c3cae42d3166cd6b2890a23a558da4daf54a3bc211280c1224f3c84ee7b7ef6f`
24	2026-08-25 00:09:39	BTC	963 926	`5f6967eb6ccc6f116907bbb60f2a03c15d92f3889e2c9922a1cc97e2ec121e2bbb4a5987edafb2345c92f233317b9a4406549e551b0db671b5c70d1eb314d6f7`
25	2026-08-25 06:06:47	ETH	25 830 312	`0x0d0449b0dbcf6d6d2bc757d19ddf2f96b9d66b5aedc383a87db522bc17a06aef4ddbfc434b6ce7e38c16ad10e942a08b380bf6ef9edd9f2a1f6dd76e648c47b0`
26	2026-08-25 12:17:30	BTC	964 001	`bfce3131633a2c182e6c778943c9c2722b1ad6ece48d77a9fdafe7c66f376f4df966ba3bfe2a7e7dab2a96a94dbfa8f87f26f5f61e0ccbf69f3f3283329e8bcb`
27	2026-08-25 18:06:47	ETH	25 833 895	`0x06677017b056cc76ad2281e13189db6c777a0c114887f290cf0112778727f58e64fac946393ef03af4fb94c4cab0726e097c02fd96a7949c26af85506a9c1604`
28	2026-08-30 00:12:02	BTC	964 643	`5db8c5cf716d887b95cef7bf5b70507e26876a60697c40c9f5db346d5eec51105dafd8bff4c69f9f69af02bd448cad0fcd0e676a4424b652da58e223a2502be3`
29	2026-08-30 06:06:47	ETH	25 866 185	`0x16875c9d18689d6f1d9bddaf791d798850df1b74e5b25aa9d2530af3746a049f62abaf70330faad1a5d1c2203c604d25c06f3de84513577e1667c77ca6c7cfdc`

5. What each anchor covers

Same numbering as section 4.

#	Anchored artefact	Notes
1	IIC v6 single-file generator, IPFS release	IPFS CID QmXXvokcgvtjqCsyWynRkuhco6YWhPqdDfRevee2zbLBsC. Reference file SHA-256 ed70cef5a086e264386ab026edaaabb8f25c378f1e3a699a2c0a72a20d53c7fb. Derived-address scheme.
2	v8.4.1 documentation bundle	Three PDF specifications, one anchor. Superseded.
3	v8.4.1 documentation bundle	Bitcoin leg of the same batch. Superseded.
4	Complete Package v8.4.1 and Wallet User Guide v1.0	Same Merkle batch. Cited in TDCommons #10394.
5	Complete Package v8.4.1 and Wallet User Guide v1.0	Ethereum leg of the same batch.
6	Complete Package v8.4.1, final UX build	Archive SHA-256 468c6bfcb140032df6abc91568bff61d300298bf86fd9a57bf78a4f0d8257fc4, 192,331 bytes. Superseded.
7	Complete Package v8.4.1, final UX build	Ethereum leg of the same batch.
8	Complete Package v8.4.1, audit-hardened	Archive SHA-256 a26ccb19f13301bfb04cffbce9dfbd5b73534181867c736b3d8d1f40e1a9289f, 194,022 bytes.
9	Complete Package v8.4.1, audit-hardened	Ethereum leg of the same batch.
10	Threat Model v8.4.1	Superseded by the v9.0 documentation set.
11	Threat Model v8.4.1	Bitcoin leg of the same batch.
12	v9.0 documentation set, TDCommons #10795 and #10796, initial v9.0 package build	Seven fingerprints in one batch.
13	v9.0 documentation set, TDCommons #10795 and #10796, initial v9.0 package build	Ethereum leg of the same batch.
14	Complete Package v9.0, corrected build	Archive SHA-256 7b2dcffde07a29975097cff3d0246ea460762630dd8037798f7644e063818afd.
15	Complete Package v9.0, corrected build	Bitcoin leg of the same batch.
16	TDCommons defensive publication #11121	Document SHA-256 966b24d856e478dc378a6a0bbe58246d0d693c447bdfda9fd8e96ffb91ba22db.
17	TDCommons defensive publication #11121	Bitcoin leg of the same batch.
18	IIC v10.0 release	
19	IIC v10.0 release	Ethereum leg of the same batch.
20	IIC v10.0.1 release	Ethereum leg precedes the Bitcoin leg by six hours; the pair spans two calendar days.
21	IIC v10.0.1 release	
22	Complete Package v10.0.1 manifest, site root r8, and the archive itself	Three leaves in one tree: SHA256SUMS 265e7eafa6ecff90cd2d3e878bd57efd10c225eacd8014a9febc504ca625a67b, site root 818ceb7f4e9d2528b0e7e35453749ef7c118665a10334ccce6ab23e9d6f513b5, archive 0febc79c46d969921152474b0b423add64bbc4523c3ee7d2ea7c3527efb996a0.
23	Complete Package v10.0.1 manifest, site root r8, and the archive itself	Ethereum leg of the same batch.
24	Site root r9	5d54149b91edd80d0728273b1c2d945e00a4a27df94f7d6a0a3a689553a401ce.
25	Site root r9	Ethereum leg of the same batch.

#	Anchored artefact	Notes
26	Site roots r10 and r11	Two leaves in one tree: r10 0e422aaa2194ccd2f65da98ed45581af97047a616b9a44539481ed81d09a4cc6, r11 ebc31fa5308bd0ebfe3b0f4b71ae33ef2411d1364a7ad3f2c2c2f7fb5bce03d7.
27	Site roots r10 and r11	Ethereum leg of the same batch.
28	Site root r12	883f9f8c9a69a3b54ad80a9b4f161d9c76c64a106e168bf5fa39b7de2d669cdc. Current deployed state.
29	Site root r12	Ethereum leg of the same batch.

6. Inclusion proofs

Because a transaction commits only a Merkle root, and a root may cover several documents, reading the root from the chain is not sufficient to prove that a particular file is under it. The link is the inclusion path: the sibling digests that allow the root to be recomputed from the leaf upward.

Nine inclusion proofs have been recomputed node by node and are published at internetidentitycard.com/proofs/ as JSON files. Each was verified to reach the root written on chain.

Leaf	Chain	Depth	Committed root
Complete Package v10.0.1 archive	BTC	13	`6f6ddc8eb8619b4ee7693e6d275ef8149ce4792cee6499b33d025c8f6cef423c`
Site root r9	BTC	12	`bb4a5987edafb2345c92f233317b9a4406549e551b0db671b5c70d1eb314d6f7`
Site root r9	ETH	10	`4ddbfc434b6ce7e38c16ad10e942a08b380bf6ef9edd9f2a1f6dd76e648c47b0`
Site root r10	BTC	12	`f966ba3bfe2a7e7dab2a96a94dbfa8f87f26f5f61e0ccbf69f3f3283329e8bcb`
Site root r10	ETH	13	`64fac946393ef03af4fb94c4cab0726e097c02fd96a7949c26af85506a9c1604`
Site root r11	BTC	12	`f966ba3bfe2a7e7dab2a96a94dbfa8f87f26f5f61e0ccbf69f3f3283329e8bcb`
Site root r11	ETH	13	`64fac946393ef03af4fb94c4cab0726e097c02fd96a7949c26af85506a9c1604`
Site root r12	BTC	9	`5dafd8bff4c69f9f69af02bd448cad0fcd0e676a4424b652da58e223a2502be3`
Site root r12	ETH	9	`62abaf70330faad1a5d1c2203c604d25c06f3de84513577e1667c77ca6c7cfdc`

Site roots r10 and r11 share a root on each chain: they are two leaves of the same tree. This is the batching described in section 2, shown concretely.

6.1 Tree construction

Each internal node satisfies:

```
node = SHA-256( ASCII(left digest, lowercase hex) || ASCII(right digest, lowercase hex) )
```

The two child digests are concatenated **as hexadecimal text**, not as raw bytes. This was determined empirically: raw-byte concatenation fails at every node. The detail appears on no certificate and is recorded here because without it a proof cannot be replayed.

7. Verification procedures

Two schemes appear in this register. Both are given in full. Neither was documented in version 1.0.

7.1 The 2019 scheme - derived address

The Merkle root is treated as a 32-byte secp256k1 private key; the corresponding address is funded with a small transaction. The root is not written on chain and must be reconstructed. Entry 1 uses this scheme: the transaction has no OP_RETURN, and pays 550 satoshi to address 1uHbSPuRqg2pUWSUY3UruvjELLEuVwW5f.

Step	Action
1	Compute the SHA-256 of the May 2019 release file. Confirm it appears as a leaf in the preserved Merkle proof.
2	Reconstruct the tree bottom-up from the proof and confirm the root equals 5bced000f3226ddb6460dc9e85a1f1198604d4f74c12abf6b35361678919a593.
3	Treat that root as a private key. Apply standard Bitcoin address derivation: secp256k1 public key, SHA-256, RIPEMD-160, Base58Check with mainnet prefix 0x00.
4	Confirm the result is 1uHbSPuRqg2pUWSUY3UruvjELLEuVwW5f.
5	Look up that address on any block explorer. Confirm transaction 929bb2d8... credited it on 13 May 2019 at 00:13:27 UTC.

7.2 The 2026 scheme - root written on chain

Entries 2 to 29 write the Merkle root directly. On Bitcoin it is an OP_RETURN carrying OP_PUSHBYTES_32 followed by the root. On Ethereum it is calldata to contract 0x7413cb703c754b6c5951da49c85ab2f3ddb6792a, selector 0x08891a32 followed by the root. This is simpler than the 2019 scheme: the root is readable without any derivation.

Step	Action
1	Compute the SHA-256 of the file. This is the leaf.
2	Open the inclusion proof and confirm the leaf appears at maximum depth.
3	Walk upward. At each node compute SHA-256 of the two child digests concatenated as hexadecimal text, and confirm it equals the node digest.
4	Confirm the computed root equals the root stated on the certificate.
5	Read the root from the transaction: Bitcoin, in the OP_RETURN output; Ethereum, in the calldata after the four-byte selector.
6	Take the block time in UTC. Never the time displayed on the certificate. See section 8.

7.3 Reference implementation

The following replays an inclusion proof and prints the root and the leaf. It uses only the Python standard library.

```
import json, hashlib, sys
node = json.load(open(sys.argv[1]))
root = node["sha256"]
while node.get("leftSha256"):
    pair = node["leftSha256"] + node["rightSha256"]
    assert hashlib.sha256(pair.encode()).hexdigest() == node["sha256"]
    child = node.get("left") or node.get("right")
    if child is None: break
    node = child
print("root", root)
print("leaf", node["sha256"])
```

Applied to the r12 Bitcoin proof, this prints root 5dafd8bf... and leaf 883f9f8c..., which is the site root of the deployed revision. The root is the value in the OP_RETURN of Bitcoin block 964 643. That single command closes the chain from file to block.

8. Corrections and known defects

8.1 Corrected in this version

Version 1.0 gave four anchor times two hours ahead of the true block time, under a column headed "Anchored at (UTC)". The cause is mechanical: the timestamping service displays local time, and those displayed times were transcribed faithfully. The transaction identifiers and calendar dates were correct; only the clock was wrong.

Entry	Stated in v1.0	True block time (UTC)
Anchor #3, Bitcoin	12 June 2026, 02:22:00	12 June 2026, 00:22:00
Anchor #3, Ethereum	12 June 2026, 08:07:23	12 June 2026, 06:07:23
Anchor #4, Bitcoin	17 June 2026, 02:32:47	17 June 2026, 00:32:47
Anchor #4, Ethereum	17 June 2026, 08:06:47	17 June 2026, 06:06:47

The operating rule adopted as a result: never transcribe a time from a timestamping certificate. Read the block time from the chain.

8.2 A defect that cannot be corrected

The same error appears in TDCommons defensive publication #10394, published 8 June 2026 under a Creative Commons licence. It states the Complete Package v8.4.1 anchors as 7 June 2026 at 02:12:14 UTC (Bitcoin) and 08:07:11 UTC (Ethereum). The true block times are 00:12:14 and 06:07:11. The error appears three times in that document: in the header table, in section 4.4 and in section 9.

A defensive publication is a permanent public record and cannot be amended. The disclosure is made here rather than left to be found: the transaction identifiers and the calendar date in #10394 are correct, so the anchors verify; only the stated clock times are two hours ahead. Entries 4 and 5 of this register give the corrected values.

8.3 Truncated identifiers in a published document

TDCommons #11121, published 23 July 2026, gives four transaction identifiers as eight-character prefixes. A prefix is not searchable on any block explorer, so that publication is not self-sufficient for verification. Its stated dates are correct. The full identifiers are entries 12 to 15 of this register.

8.4 Version 1.0 metadata

Version 1.0 carried reference IIC-BAR-2026-06-14 and a date of issue of 14 June 2026, yet documented an anchor dated 17 June 2026. It was extended after issue without a version increment. Recorded here for completeness.

9. Supporting registrations

Listed for context. None is established by the anchors in this record; each rests on its own register.

Right or record	Identifier	Status
UK trademark, INTERNET IDENTITY CARD	UK00003166480	Registered, filed 25 May 2016, renewed through 2036
French trademark	FR 4071419	Lapsed, not renewed at the 2024 term
US copyright	TX 8-508-373	Registered 9 November 2017. Work completed and first published 2014. Type of work: text; authorship claimed: text and photographs. Claimant: Michael Benaudis
UK company	09168431	Incorporated 8 August 2014
ICO data controller	ZA457585	Registered
D-U-N-S	220301971	Assigned
ORCID	0009-0008-7763-1691	Active
Zenodo DOI	10.5281/zenodo.21738537	Published
IETF Internet-Draft	draft-benaudis-iic-credential-00	Submitted
TDCommons defensive publications	10079, 10167, 10394, 10795, 10796, 11121	Published, CC BY 4.0

10. Sources and reproduction

Every claim in this record can be checked without contacting the issuer and without an account, permission or fee.

Resource	Location
Bitcoin explorer	`mempool.space/tx/{txid} - blockstream.info/tx/{txid}`
Ethereum explorer	`etherscan.io/tx/{txid}`
Bitcoin API used here	`mempool.space/api/tx/{txid}`
Ethereum RPC used here	`ethereum-rpc.publicnode.com (eth_getTransactionByHash, eth_getBlockByNumber)`
Ethereum anchoring contract	`0x7413cb703c754b6c5951da49c85ab2f3ddb6792a`
Inclusion proofs	`internetidentitycard.com/proofs/`
Site manifest and integrity page	`internetidentitycard.com/SITE-SHA256SUMS - /integrity-public`
Anchor register, public	`internetidentitycard.com/verify`
2019 IPFS release	`QmXXvokcgvtjqCsyWynRkuhco6YWhPqdDfRevee2zbLBsC`